Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A computer-implemented method of validating a document structured as an ordered tree having labeled elements, known to conform to varying element types in accordance with a first schema, with respect to a second schema ~~with respect to a second schema given the knowledge that the document is valid with respect to a first schema~~, said method comprising:

~~preprocessing said first schema and said second schema to compute information to assist in the validation of the document with respect to said second schema~~

preprocessing the first and second schemas to identify subsumed type-pairs, of the form type1-type2 where type1 is a type defined in the first schema and type2 is a type defined in the second schema, the pairs indicating that an element's content that conforms to the first type, type1, will also conform to the second type, type2; and

identifying subsumed element tag-type-pairs, of the form tag-type1-type2, where tag is an element name, type1 is a type in the first schema and type2 is a type in the second schema; it indicates that if the specified element tag appears in a document conforming to the first schema with content of type1, then this element tag can validly appear in a document conforming to the second schema and its content will conform to type type2.

2. (Original) The method of claim 1 wherein the step of preprocessing comprises comparing said first schema and said second schema to determine relationships between said first schema and said second schema.

3. (Original) The method of claim 2 wherein the step of preprocessing comprises determining relationships between types defined in said first schema and said second schema.

4. (Original) The method of claim 3 wherein a type defined in said second schema is identified as a subsumed type if a relationship exists between a type in said first schema and said type in said second schema such that portions of a document that are valid with respect to said type in said first schema are also valid with respect to said type in said second schema.

5. (Original) The method of claim 3 wherein a type defined in said second schema is identified as a disjointed type with respect to a type in said first schema if a relationship exists between said type in said first schema and said type in said second schema such that portions of a document that are valid with respect to said type in said first schema are not valid with respect to said type in said second schema.

6. (Original) The method of claim 3 wherein a type defined in said second schema is identified as an intersecting type if a relationship exists between a type in said first schema and said type in said second schema such that some portions of a document that are valid with respect to said type in said first schema are valid with respect to said type in said second schema and some portions of a document that valid with respect to said type in said first schema are invalid with respect to said type in said second schema.

7. (Original) The method of claim 3 wherein said relationships between said types defined in said first schema and said second schema are used to determine whether said document is valid or invalid in said second schema.

8. (Original) The method of claim 5 wherein a document is determined to be invalid with respect to said second schema if said document contains types that are identified as disjointed.

9. (Original) The method of claim 4 wherein any portions of a document that are of a subsumed type are immediately accepted as valid with respect to said second schema.

10. (Original) The method of claim 3 wherein types assigned to document elements while validating with respect to said first schema are used to validate the document with respect to said second schema.

11. (Original) The method of claim 10 wherein said types assigned to said document elements while validating with respect to said first schema are provided with said document.

12. (Original) The method of claim 10 wherein said types assigned to the document elements during validation with respect to said first schema are computed while validating said document with respect to said second schema.

13. (Original) The method of claim 10 wherein a document is deemed invalid if a type assigned to a document element during validation with respect to said first schema is in a disjoint relationship with a type in said second schema.

14. (Original) The method of claim 10 wherein a document element is deemed valid if said document element is assigned a type during validation with respect to said first schema that is in a subsumed type relationship with a type in said second schema.

15. (Original) The method of claim 10 wherein said computed information and said assignment of types when said document is validated with respect to said first schema are used to determine portions of said document that are to be validated according to said second schema.

16. (Original) The method of claim 15 wherein a portion of said document is validated according to said second schema if said type assigned to said portion of said document during validation with respect to said first schema is in an intersecting relationship with a type in said second schema.

17. (Original) The method of claim 16 wherein the preprocessing comprises developing an automaton from a first type in said first schema and a second type in said second schema that are in an intersecting relationship to determine if a portion of said document that is assigned said first type during validation with respect to said first schema is valid with respect to said second type during validation with respect to said second schema.

18. (Original) The method of claim 1 wherein said first schema and said second schema are one of a regular expression, document type definition, finite state automata, XML schema and tree automata.

19. (Currently amended) The method of claim 1 wherein said document is ~~one of a string, data set,~~ an XML document~~, ordered tree and tree~~.

20. (Original) The method of claim 3 wherein said types are one of states in a finite state automaton, element type declarations in an XML schema, programming language types and states in a tree automaton.

21. (Original) The method of claim 1 comprising examining said document and determining if any portions of said document have been modified subsequent to said document being validated in said first schema and limiting portions of the document to be validated based upon which portions of said document have been modified.

22. (Original) The method of claim 21 comprising identifying elements of said document that have been inserted and examining any elements that have been inserted to determine if they are valid with respect to said second schema.

23. (Original) The method of claim 21 comprising identifying elements of said document that have been renamed and using a determined relationship between said renamed elements and said element prior to being renamed when validating said renamed elements with respect to said second schema.

24. (Original) The method of claim 21 comprising identifying elements of said document that have been deleted and ignoring any deleted elements when validating said document with respect to said second schema.

25. (Original) The method of claim 21 wherein the information computed comprises a set of modification specifications such that a modification specified by said set of modification specifications results in the document being valid according to said second schema.

26. (Original) The method of claim 21 wherein the information computed comprises a set of modification specifications such that a modification specified by said set of modification specifications results in a document being invalid according to said second schema.

27. (Original) The method of claim 1 wherein said document has been modified subsequent to being validated in said first schema and wherein said first schema is the same as said second schema.

28. (Currently amended) An information handling system for validating a document structured as an ordered tree having labeled elements, known to conform to varying element types in accordance with a first schema, with respect to a second schema ~~that has been determined to be valid with respect to a first schema in accordance with a second schema~~; said information handling system comprising:

a processor configured for:

preprocessing the first and second schemas to identify subsumed type-pairs, of the form type1-type2 where type1 is a type defined in the first schema and type2 is a type defined in the second schema, the pairs indicating that an element's content that conforms to the first type, type1, will also conform to the second type, type2; and

identifying subsumed element tag-type-pairs, of the form tag-type1-type2, where tag is an element name, type1 is a type in the first schema and type 2 is a type in the second schema; it indicates that if the specified element tag appears in a document conforming to the first schema with content of type1, then this element tag can validly appear in a document conforming to the second schema and its content will conform to type type2.

~~for comparing said first schema to said second schema and determining whether a document determined to be valid with respect to said first schema is valid with respect to said second schema based upon said comparison~~.

29. (Currently amended) A computer-readable medium comprising computer code for executing a method of determining whether a document conforming to a first schema may be cast in a second schema without validating every element of said document in accordance with said second schema, the document being structured as an ordered tree having labeled elements, known to conform to varying element types in accordance with a first schema, with respect to a second schema, said method comprising:

~~identifying element types in the first schema that are always valid in the second schema and not examining individual elements in the document that are of the always valid element types in the second schema;~~

~~identifying element types in the first schema that are possibly invalid in the second schema and examining any individual elements of the possibly invalid element types in the document to determine if said individual elements are invalid in the second schema;~~

~~identifying element types in the first schema that are always invalid in the second schema and determining whether any always invalid element types are in said document; and~~

~~identifying the document as valid in the second schema only if no always invalid elements types and no invalid individual elements were identified in the document.~~

preprocessing the first and second schemas to identify subsumed type-pairs, of the form type1-type2 where type1 is a type defined in the first schema and type2 is a type defined in the second schema, the pairs indicating that an element's content that conforms to the first type, type1, will also conform to the second type, type2; and

identifying subsumed element tag-type-pairs, of the form tag-type1-type2, where tag is an element name, type1 is a type in the first schema and type 2 is a type in the second schema; it indicates that if the specified element tag appears in a document conforming to the first schema with content of type1, then this element tag can validly appear in a document conforming to the second schema and its content will conform to type type2.